XML-BASED DECISION SUPPORT SYSTEMS:
CASE STUDY FOR PORTFOLIO SELECTION

JICHANG DONG
School of Management, Graduate School, Chinese Academy of Sciences
Beijing 100080, China

HELEN S. DU
Department of Information Systems, City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong

K. K. LAI*
Department of Management Sciences, City University of Hong Kong
83, Tat Chee Avenue, Kowloon Tong, Hong Kong
mskkla@cityu.edu.hk

SHOUYANG WANG
Institute of Systems Science, Academy of Mathematics and Systems Sciences
Chinese Academy of Sciences, Beijing 100080, China
College of Business Administration, Hunan University
Changsha, Hunan 410082, China

The extensible, structural and validated nature of XML provides standard data representation for efficient data interchange among diverse information resources available on the Web. Therefore, it leads to its growing recognition in e-commerce and Internet-based information exchange. In this paper, we stress the adoption of XML technology in developing efficient and flexible Web-enabled decision support systems. Based on a case study for portfolio selection systems, we explore the design issues in applying XML to overcome the heterogeneity of data exchange and sharing of various portfolio optimization models.

Keywords: XML; portfolio selection; decision support system; open interchange.

1. Introduction
The Internet has become one of the most exciting information dissemination media in recent years. It offers many advantages over traditional media, being inexpensive and widely used as the backbone infrastructure for global information sharing and exchange. Through the World Wide Web, users can retrieve information and request

*Corresponding author.
services at few keystrokes. Web-based Decision Support Systems,\textsuperscript{6,8} Web DSS, one of the important areas of Web applications, has attracted increasing attention from researchers and organizations.

Portfolio selection, an important topic in the financial management field, is a bi-criteria optimization problem that leverages a reasonable trade-off between the expected rate of return and a degree of risk. In real-life situations, we tend to calculate solutions each time portfolio adjustments are made, so the decision and impact of adjustments can be estimated. Clearly, this would be impractical if manual calculations would have to be re-done at iteration, due to the long time delays involved. In the last few years, many Web-based mean-variance optimization decision support systems are available to support portfolio selection.\textsuperscript{5} Most of the existing Web DSS with the portfolio selection features are designed in client/server architecture. The client site is mostly developed under HTML, and can be viewed using any HTML browser.

XML, an extensible markup language for structured data, can be used to encode information with human- and machine-readable semantics.\textsuperscript{3,12} XML-enabled data can use document type definition (DTD) to define their structures and set the constraints on them. The DTD file allows the user to specify a set of tags, the order of tags, and attributes associated with each tag. Both HTML and XML are derived from SGML, a meta-language that has existed for over 20 years. However, XML has many benefits that cannot be fully provided by HTML, such as simplicity, extensibility, openness and interoperability. These characteristics can satisfy some requirements for modeling tools and data exchange in today’s Internet age.

In this paper, we present XML-based DSS schemes based on an integrated solution for portfolio selection, as compared to the traditional HTML-based approach. First, we introduce our integrated model to portfolio selection. Then, we expand our discussion on applying XML to the portfolio selection framework provided and explore the issue on data heterogeneity and sharing of decision models raised in the adoption of XML technology. Finally, we compare the XML-based method with the HTML-based system WPSS (http://madis1.amss.ac.cn) we developed.

2. Framework of Portfolio Selection

Portfolio selection should be considered as a process that includes several related steps, rather than just solving an optimization problem. Here, we provide an integrated model that involves the combination of asset allocation, securities analysis, securities selection, portfolio optimization, and rebalancing processes in building an efficient and flexible Web DSS for portfolio selection.

\textit{Asset Allocation.} This is the first stage in which the decision-maker should determine the optimal mix of assets (such as large cap stocks, small cap stocks, international stocks, bonds and cash) that will comprise his or her investments.
Asset Analysis. This is an important stage, which enables a decision-maker to conduct an analysis of securities that may be under consideration for inclusion in a portfolio.

Asset Selection. This stage may use techniques such as profiles, where security attributes from the previous stage are examined in advance of the regular selection process. Any securities, which do not meet pre-set criteria (such as average return, return volatility and Sharpe Ratio requirements) are eliminated.

Portfolio Optimization. Portfolio optimization is a major stage in portfolio selection. This ensures optimal risk-adjusted returns by analyzing the portfolio and managing the assets. Some comparative optimization models can be considered in this stage, such as a mean variance optimization (MVO) model taking transaction costs (not fixed) into consideration.\textsuperscript{1,4,10} As an important part of the integrated solution provided, portfolio optimization can be further broken down into sub-processes including Constraints (on portfolio assets), Asset return (profile), Asset volatility (profile), Correlation (of portfolio assets) and Optimization. These steps allow decision-makers to assess how close to the optimum their portfolios are performing.

Rebalancing. In this stage, decision-makers apply their knowledge and experience to balance and make other adjustments to the portfolio by adding or disposing of securities. This adjustment can help to achieve balance among the securities selected. There should not be too high a proportion of high-risk securities because failures of several of these securities could be dangerous to the value of the investment. On the other hand, if the portfolio selected is too conservative, the expected return may be too low. Furthermore, the balance of the size of investments in any individual security is also important, because the commitment of a high proportion of the investment capital to a few securities can be catastrophic if more than one fails. Also, too many long-term investments may cause financing or cash flow problems.

3. XML-Based 3-Tier Scheme for Portfolio Selection

XML’s power lies in its flexibility and simplicity. Derived from a philosophy that uses a standard data representation format, this frees XML from binding to any particular script language, authoring tool or delivery engine.\textsuperscript{2}

In the following, we provide an XML-based three-tier decision support system scheme for portfolio selection depending on the framework we proposed compared with the HTML-based system.

In Fig. 1, we present an XML-based three-tier structure following the integrated portfolio selection introduced earlier. Our main goal is to provide a user-friendly interface with interactive feedback and ease of use with rapid response. This is critical to all users, especially executive decision-makers.
The three tiers of our solution are the clients, the middleware servers, and the back-end applications. The Java client, part of the middle-tier and back-end systems, can be written in Java and run on multiple platforms without modification.

**Clients.** It displays results to the users and allows them to interact with the system to arrive at satisfactory solutions. The user interface is used by decision-makers to input data and decision preferences, and to retrieve answers from related services by the clients. The clients can also be run as applets in Java-enabled browsers.

**Middle Tier.** The middle tier mainly includes standard Web servers, application logic, and decision switchers. A portfolio selection framework is used to aid the users to expedite their decision-making processes in a more organized fashion. The decision switcher component is responsible for converting the received results in XML format and returning them to the gateway tools, relaying requests from application logic to agents and controls the flow of results in the reverse direction. Decision switchers have a multi-thread architecture, which allows simultaneous communication with multiple agents. They are information brokers which receive specific requests from Web clients, determine the proper agent for their execution, relay requests, combine results coming from more than one agent, and provide for
session management. Moreover, some decision switchers can represent, share and manage different external portfolio optimization models in open architecture. The middle server is responsible for receiving requests from the clients, accessing the data source, authenticating users and performing session management, formatting the received results into XML if necessary and returning them to the clients, as well as dealing with the decision models.

Back-End Applications. The third tier consists of agents, data sources, and portfolio optimization model program (different to the external shared models). Data sources include legacy systems such as conventional RDBMS (e.g. SQL Server and Oracle) and can be used as a repository of data (including the data from Web sites) for all model programs and search requests. Agents are used to control the interaction with back-end applications. Access to the data sources is accomplished through well-established protocols such as Microsoft’s ODBC or OLE DB, JDBC, and ADO. Agents can also operate like typical Web spiders. The role of an agent is to establish a two-way connection with the back-end applications, and server requests arriving from one more decision switchers or application logic.

4. Interactions and Implementation

On the client side, the Web browser and Java applet handle the user interface and the presentation logic. The Web server gets all HTTP requests from the Web user and propagates the requests to the application server that implements the logic of all the services for portfolio selection. Communication between the Web server and the application logic can be achieved through the CGI, ASP or other gateway tools. An application server sends a query to the agents and gets the result set. The decision switchers can format the result into XML if necessary and provide the XML document to the client. Decision switchers operate behind classical Web server using interfaces such as CGI or other gateway tools. Decision switchers map meta-information as well as the actual data to a proposed XML DTD. The DTD describes how certain objects are described in XML using a specific tag set, and thus allows the client to validate response against the DTD. The server script simply fetches the requested objects from the database and transforms the python objects into XML-tagged output. XML streams are transmitted to clients using regular HTTP dialogues, and XML is received and processed by a Java applet. After the applet’s invocation, communication with the server is still performed using the HTTP protocol but the HTTP response message always contains XML code in its body. The Java client receives an XML page, parses it and generates Java objects that are then graphically displayed by the applet. At the third tier, agents handle individual information sources such as RDBMS, and the schemata model of each individual source mapped to a generic representation known to both agents and decision switchers. Portfolio optimization models can be maintained in a separate database. If the external shared portfolio models are required, some decision switchers can be used to make the system work well.
It is clear that we need to develop a better interface to the middle server that allows the (Java) client access to the data in a more comprehensible way. We can use an HTTP/XML-based approach, and construct a DTD for data structures. The DTD describes how certain objects are described in XML using a specific tag set, and thus allows the client to validate their response against the DTD. The server script simply fetches the requested objects from the database and transforms the python objects into XML-tagged output.

The Java applet is based on publicly available XML software for Java. We can develop the interface between the applet and its XML components using simple API for XML (SAX), which connects the XML parser with the main application via a specific method XML-parser. We use the document object model (DOM) to provide standard objects for accessing parsed XML documents. Also, SVG can be of great help in displaying all kinds of charts to help in the decision-making processes of portfolio selection. SVG is the World Wide Web Consortium’s (W3C’s) new graphics language based on XML and is expected to become the standard for web-based graphics. SVG is an excellent application to centralize different data sources in a structured and open way, while allowing for rich graphics.

In the interaction process, a registered user can log in and pick an existing session or create a new session of his/her own preference, such as level of user protection, access rights and level of complexity. User protection and session management are maintained in a security system interface level following with asset allocation, securities analysis, securities selection, portfolio optimization, and rebalancing modules.

4.1. XML for heterogeneous data in decision process

In the portfolio selection process, various types of data may be used. For example, stock market data, bond market data, and other data from Web pages in the financial market. These data may be saved in heterogeneous databases according to their usage in the portfolio selection process. Especially, heterogeneous data sources might be available for posting a single request constantly. In this 3-tier structure we can deal with the problem of heterogeneous databases in a portfolio selection process through the agents and the decision switchers. Agents are used to control the interaction with specific data sources. Access to the data sources is accomplished through well-established protocols such as Microsoft’s ODBC or OLE DB, JDBC, and ADO. Agents can also operate like typical Web spiders. The role of an agent is to establish a two-way connection with a specified data source, and server requests arriving from one more decision switchers.

To ensure the generality of the agent, the problem of different data types and different data-type inner representations have to be dealt with. For example, the data may be collected from some Web sites, so we need to define the structure of the data that we would like to receive by using an XML DTD file. This DTD file will then be referenced by switchers to translate source data into an XML format.
We can define a simple XML DTD to specify the tags and structure of the source data retrieved from the Web as shown in Fig. 2.

Since XML has the characteristics of being self-describing and inherently hierarchical, it provides both the richness to describe heterogenous source data exchanged in the portfolio selection process, and the flexibility to accommodate the diversity of intelligent agent access. By adopting an XML code, the processing load moves from the server to the Web clients. As XML is used increasingly on the Web, XML-based content will make the Web virtually fully accessible and scalable to intelligent agents and other automated processes.

4.2. Open interchange of decision models

Portfolio optimization model is a key component in the portfolio selection process, and the modeling process is knowledge-intensive and time-consuming. There has been much research on sharing models that may be in different environments. In order to support modeling processes and related activities, there has been a lot of research on modeling environments (ME). To implement such a ME, a conceptual modeling framework is important to represent and manage decision models.
In general, a closed architecture for interchange of models and data may produce a tangle of import and export point translators. An open exchange standard improves the shortcomings of closed architecture. Having a standard syntax for creating and exchanging data structures is obviously important for this type of integration. XML provides such a framework for describing the syntax. The distinctive characteristics of XML can satisfy some of the new requirements for modeling tools and technologies in the age of the Internet, such as simplicity, extensibility, interoperability and openness. There are many approaches and systems that involve XML in the standardization of data exchange and presentation, such as CML (Chemical Markup Language), MathML (Mathematical Markup Language), and ebXML, reflects the growing requirements and importance of XML.

We can adopt XML as a meta-language and construct a Web-based structure (shown in Fig. 3) for the open interchange of portfolio optimization models in different environments. An XML-based language (a central switcher in Fig. 3), formally defined by a simple XML DTD, for the representation, sharing and management of different portfolio optimization models in open architecture, is required. The language is able to allow applications to access models obtained from multiple sources without having to deal with individual differences between those sources. The modeling environments are based on the Web for sharing modeling knowledge. In this open architecture, we can use Markov decision process or fuzzy technology to optimize load balancing for multi-computer systems.

Du et al.\textsuperscript{7} give a generalized XML DTD definition to describe heterogeneous portfolio optimization models. As shown in Fig. 4, a model has its own unique name and pre-defined format for parameters passed in and out. The type of parameters for a model, in this case, can be of “SI” for system-generated input; “UI” for user input; or “O” for output parameters.

Following the standards defined in the DTD file, we can then further specify each individual model regardless of what language or system it uses. In the case of MVO model that we introduce earlier, its profile can be specified as shown in Fig. 5.

![Figure 3. Architecture for model sharing based on an XML-standard.](image-url)
Fig. 4. XML DTD for portfolio optimization model.

```xml
<!ELEMENT model (name, parameter+)>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT parameter (name, type)>  
  <!ATTLIST parameter ptype (SI|UI|O) #REQUIRED>  
<!ELEMENT type (#PCDATA)>  

<model>  
  <name>GetMVO</name>  
  <parameter ptype="SI">  
    <name>MeanReturn</name>  
    <type>float</type>  
    <name>Convariance</name>  
    <type>float</type>  
  </parameter>  
  <parameter ptype="UI">  
    <name>MinReturn</name>  
    <type>float</type>  
    <name>RiskLevel</name>  
    <type>RiskLevel_type</type>  
  </parameter>  
  <parameter ptype="O">  
    <name>OptimalMVO</name>  
    <type>float</type>  
  </parameter>  
</model>
```

Fig. 5. Sample profile of MVO model.
5. Comparisons to HTML-Based Portfolio System WPSS

Similar to the HTML-based system WPSS (Web-based portfolio selection system, http://madis1.amss.ac.cn) we developed earlier, in this paper, we also include an integrated framework (as discussed in Sec. 2) for portfolio selection adaptable to the needs of financial organizations and individual investors. Portfolio selection process, hence, becomes more flexible and organized since decision-makers can interactively select and change their initial investment, time horizon, yearly contribution, tax impact, and the amount of risk they are willing to take according their own preference. However, comparing with the previous HTML-based portfolio selection, the current XML-based design has some key advantages over the other as discussed in the following.

One of the most important benefits of our method is the re-distribution of processing load from the server to the Web clients. In WPSS, the inner-structure of the retrieved information was undervalued while the scope of the relevant semantic content was restricted to the server-side. Such limitations can be overcome by XML, which caters for a more structured handling of the exchanged information. The client can do many things that were originally done by server, which can improve the performance of the system and save time during which programmers might otherwise have to update the system.

The second is that the Web client (Java applet) may present different views of the same data to different users due to its intelligence in our method. Because HTML was not designed to support advanced user-interfaces and views, in WPSS developers have to circumvent HTML’s limitations using proprietary client extensions, such as JavaScript, ActiveX, or other task-specific plug-ins. Ironically, such approaches diminish the Internet’s biggest advantages of universal accessibility and environmental independence. XML allow us to exploit the Web in ways that HTML cannot, as rich data delivered to the desktop can be computed, and presented locally. As we know, we need to provide different views of data to help users analyze all kinds of data and market information sometimes in the process of portfolio selection.

The third is that our method has structural validity checking mechanism because of the XML’s benefit. In WPSS, HTML designs are inherently more error-prone and require the developer to incorporate error management routines to identify invalid data. In the portfolio selection process, data validation is very important to guarantee efficient decision-making. Our method incorporates data validation through the DTD of an XML message and XML method is not complex. What is more, by making use of XML, the programmer can save lots of times that might previously have been used to develop a validity-checking mechanism.

In the portfolio selection process, various types of data may be used. For example, stock market data, bond market data, and other data from Web pages in the financial market. These data may be saved in heterogeneous databases according to their usage in the portfolio selection process. Most Web pages are an unstructured
collection of static and dynamically generated HTML pages that are difficult for external applications, like intelligent agents, to extract useful information. In WPSS, the HTML approach to data collection is cumbersome because HTML pages routinely include non-informative data such as graphics. In contrast, XML method is an open standard that is self-describing, which simplifies intelligent agent access to XML documents and applications. An XML method makes the design of the system more convenient and easy, and makes the interaction of system more efficient.

Last, we can adopt XML as a meta-language and construct a Web-based structure for open interchange of portfolio optimization models in different environments conveniently and easily. In a distributed modeling environment, the highly structured delivery of data of XML enables open interchange between servers and clients, and potentially between servers themselves. WPSS has HTML-based modeling environments, and the model manipulation and output generation is performed on the server side.

6. Conclusions

Nowadays, most of the existing Web DSS for portfolio selection are HTML-based systems with the characteristic that the server supports the construction of specific DSS and the execution of any specific DSS. However, with the emergence of XML and rapid advance in Web technologies, applying XML to provide more efficient and flexible portfolio selection process on Web DSS becomes more appropriate and beneficial.

In this paper, we discuss XML and its application to the Web-enabled information systems. An XML-based scheme following our proposed portfolio selection model for Web DSS is presented. We provide design and discussion on the adoption of XML for heterogeneous data sources as well as sharing of various decision models in portfolio selection process. Comparing to the HTML-based system (e.g., WPSS), we recognize that the use of XML for constructing a 3-tier portfolio selection system can provide openness, inter-operability, and flexibility.

Although XML offers many benefits to Web applications including portfolio selection DSS, it is also important to be aware that many of its benefits may at the same time impose some weakness. For instance, XML’s openness leads to the problem of the lack of security; its tree-like hierarchical data structure represents the real-world objects naturally, but creates difficulty in communicating with existing databases that are relational or hierarchical.

Acknowledgement

This paper is supported by NSFC, CAS, GSCAS (project yzj200307) and the City University of Hong Kong (project 7100289).
References